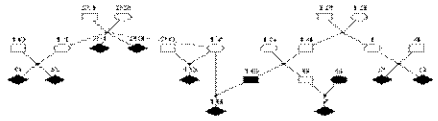
4.3 BIGGER PEDIGREES
4.3.1 Elston-Stewart algorithm

- Similar ideas underlie pedigree peeling. We don't have Markov chain, but, for pedigrees with no loops, we do have that, conditionally on the, genotype of each individual, data ``above'' is independent of data ``below''.
- Note ``above B'' means connected to B through B's parent(s), and ``below B'' means connected to B through B's offspring (and hence also spouse(s). It concerns the pedigree graph, not chronology.
- Using functions R_i(g) = P(data below | G_i =g) and
  R*_i(g) = P(data above, G_i=g)
  we can form similar equations to 4.2.4, and compute the likelihood.
- BUT these are phased multilocus genotypes, so computation is now linear in pedigree size but exponential in number of loci.
- Methods can be extended to pedigrees with loops, but this is even more computationally intensive: computation is then exponential in the number of individuals who must be considered jointly in order to cut the pedigree into independent parts (the *cutset*).

Peeling a zeroloop pedigree



- Pedigree without loops. Shaded individuals are those for whom phenotypic data are assumed to be available.
- We can work through the pedigree, accumulating the total probability of observed data (see 4.2.4) conditioning on the genotypes of the cutset individual.
- Using joint functions R* and conditional functions R we can work both up and down the pedigree.
- Packages using this approach include LIPED, FASTLINK, VITESSE

4.3.2 IBD and risk probabilities

- Note that the Baum algorithm, for small pedigrees gives us the probability of Sv(j) given all the data Y on the pedigree.
- Q_j(s) = P(Sv(j) =s | Y) is proportional to
  P(Y^(j-1),Sv(j)=s) P(Yv(j) |Sv(j)=s) P(Yv(j+1),…., Yv(*L*) |Sv(j)=s)
- The first term is R*_j(s) of 4.2.4, the second is a single-locus data probability, and the third is a probability arising in the reverse form of the Baum algorithm (see 4.2.6), We will call this one R_j(s).
- So Q_j(s) is proportional to R*_j(s) P(Yv(j)|Sv(j)=s) R_j(s) and the Baum algorithm gives both R* and R and hence probabilities Q.

- Now on a pedigree, using the R and R* defined in 4.3.1, the risk probability Q_i(g) = P(G_i=g |Y) is proportional to
  P(data above i , G_i=g) P(Y_i | G_i=g) P(data below i | G_i =g)
  which is R*_i(g) P(Y_i | G_i=g) R_i(g).
- Here, pedigree peeling gives us the risk probabilities for individuals i exactly as the Baum algorithm gives us probabilities for Sv(j) and hence ibd, by ``peeling'' along the chromosome.

4.3.3 Monte Carlo given trait data: SIMLINK

- In early 1980s, often we had trait data; trait data had simple known models (e.g. dominant, recessive).
- Marker maps were just starting; marker typing was expensive.
- How to persuade NIH that mapping is feasible and hence to fund it?
- We can simulate trait and marker data jointly quite easily, assuming some trait model, but these simulated trait data are not as observed.
- But it would be much better to simulate what marker data would look like conditional on the trait data. Ploughman and Boehnke solved this problem and implemented it in the program SIMLINK:
  1) Peel up at the trait locus, saving the partial sums.
  2) Assign founder types at the top, in accordance with trait model and upward peeling.
  3) Simulate back down at the trait locus, using the saved partial sums.
  4) Simulate at marker loci linked to the trait, at some assumed marker allele freq and recombination fraction.
  5) Do the lod score for each simulated data set.
  6) Compute an empirical Elod. If this Elod is ``big enough'' then this indicates that the trait data are sufficient for marker typing to be worth doing.
- For some years, NIH required SIMLINK evidence in proposals.

4.3.4 Monte Carlo Baum and Monte Carlo EM

- As well as giving the marginal distributions Q_j(s) = P(Sv(j) =s | Y) for locus j, the Baum algorithm also provides a Monte Carlo realization from the joint distribution P({S_ij} | Y).
- The forward computation is exactly as before. The backward computation is replaced by sampling.
- First, Sv(*L*) is sampled from Q_*L*(.). Then, given a realization of (Sv(j) =s*, Sv(j+1),…,Sv(*L*)), a straightforward application of Bayes Theorem gives
  P(Sv(j-1) =s | Sv(j) =s*, Sv(j+1),…,Sv(*L*), Y)
  = P(Sv(j-1) =s | Sv(j) =s*, Y^(j-1)) which is proportional to
  P(Sv(j) =s* | Sv(j-1) =s) P(Y^(j-2), Sv(j-1) = s) P(Yv(j-1) | Sv(j-1)=s).
- The middle term here is R*_{j-1}(s); the others are simple.
- Normalizing these probabilities over 2^m values of Sv(j-1) we have the required probabilities for SV(j-1); hence we can realize Sv(j-1).
- This is done for each j= *L*, *L*-1,…,4, 3, 2 in turn, providing an overall realization {S_ij} =(Sv(1), …,Sv(*L*)) from P({S_ij} | Y).

Map estimation: Monte-Carlo EM

- An alternative to 4.2.6 is Monte-Carlo EM.
- Instead of computing the bivariate distributions of (Sv(j-1), Sv(j)), N realizations of {S_ij}, { {S_ij}^{t}, t =1,…,N },are obtained from the conditional distribution of P({S_ij} | Y) under the current parameter values.
- Monte Carlo EM simply replaces exact computation of expectations in the E-step with a Monte Carlo estimate.
- The counts of recombinants are scored exactly as in 4.2.6:
  R^{t}_{m,j-1} ~= Σ_{i male} | S^{t}_{i,j} - S^{t}_{i,j-1} |.
- A Monte Carlo estimate of R*_{m,j-1} is Σ_{t=1}^N R^{t}_{m,j-1} / N, and the new estimate of ρ_{m,j-1} is R*_{m,j-1}/M_m as before, again with analogous formulae for all intervals and both sexes.
- (Note R and R* here have nothing to do with peeling R and R*.)
- This Monte Carlo EM is readily implemented, and, like many Monte Carlo EM procedures, performs as well as the deterministic version.
- Initially, the Monte Carlo sample size N need not be large, although for the final EM steps, close to the MLE, N should be increased.

4.3.5 Sampling ibd conditional on data: using computation, Monte Carlo, or MCMC

- Identity by descent (ibd) is a function of {S_ij}.
- By computing Q_j(s) = P(Sv(j) =s | Y), j=1,…,L, (see 4.3.2) we can compute probabilities of ibd at each locus j given data Y.
- Alternatively, by sampling {S_ij} from P({S_ij} | Y) we can estimate gene ibd on pedigrees as in 4.3.4.
- However, this is only feasible on small pedigrees, since the forward computation uses the Baum algorithm.
- Instead, we can choose a random subset of meioses i, and resample Sm(i) given Y and given all the Sm(k) for k not in the subset.
- This is just a Baum and resampling algorithm for the (small) number of meioses in the subset. (All the other Sm(k) are held fixed.)
- This defines a Markov chain over the space of {S_ij} values. Subject to various conditions the equilibrium distribution is P({S_ij} | Y). So we can just keep repeating the resampling process, to get (dependent) realizations from P({S_ij} | Y). (This is MCMC.)

4.3.6 MCMC for lod scores on big pedigrees

- There are various ways to use {S_ij} to estimate lod scores for location of a trait locus given a genetic marker map.
- The simplest is the Lange-Sobel (1991) approach – implemented in the MORGAN programs lm_markers and lm_multiple.
- Let Z be trait data, and Y the marker data, and {S_ij} the inheritance patterns at the markers. Let ξ be hypothesized position of trait locus. The marker map and allele frequencies are assumed known.
- Then, given a fixed model for Y, P(Z, Y ; ξ) is proportional to P(Z | Y ; ξ) = Σ_{S_ij} P(Z | {S_ij}; ξ) P({S_ij} | Y)= E(P(Z | {S_ij}; ξ)| Y).
- So we can sample {S_ij}^{t}, t=1,…,N given marker data Y and estimate P(Z | Y; ξ) by averaging the resulting values of P(Z | {S_ij}^{t}; ξ) over the N realizations.
- By estimating the required at different hypothesized ξ, we will have a Monte Carlo estimate of the lod score curve (over ξ).
- An advantage of this particular approach is that only one sequence of MCMC realizations {S_ij}^{t}, t=1,…,N are required, since the MCMC is conditional only on Y.